Banco Pichincha - DevOps Technical Assessment

Skills required:

- GNU/Linux Advanced Administration
- Proficiency in: Python, Java, Go, Ruby or Node (any of these programming languages)
- Advanced GIT knowledge
- CI/CD pipelines knowledge is a must. (Any on-premise or cloud platform)
- Use of IaC (Infrastructure as Code) (With any desired provider)
- Kubernetes Administration
- Docker Administration

Desirable Skills:

- Cloud Computing administration (Microsoft Azure, Amazon AWS, Google Cloud Platform, etc.)

DevOps Exercise

Rules and requirements

- There is no internet usage limit
- All generated code must be versioned using a public **GitHub** Repo created by the candidate.

Requirements:

Build a microservice that must have a REST endpoint named: /DevOps (use any preferred programming language or framework).

This is the JSON payload for the endpoint. Please use a HTTP POST method:

```
{
 “message” : “This is a test”,
 “to”: “Juan Perez”,
“from”: “Rita Asturia”,
“timeToLifeSec” : 45
}
```

This is the JSON payload the endpoint must return:

```
{
“message” : “Hello Juan Perez your message will be send”
}
```

Other HTTP Methods calls must return the string “ERROR”

The EndPoint must be secured with this APIKey:

2f5ae96c-b558-4c7b-a590-a501ae1c3f6c

The APIKey must be included in HTTP Headers

In our side, we will use this command to test your endPoint

```
curl -X POST \
 -H "X-Parse-REST-API-Key: 2f5ae96c-b558-4c7b-a590-a501ae1c3f6c" \
 -H "X-JWT-KWY: ${JWT}" \
 -H "Content-Type: application/json" \
 -d '{ “message” : “This is a test”, “to”: “Juan Perez”, “from”: “Rita Asturia”, “timeToLifeSec” : 45 }' \
 https://${HOST}/DevOps
```

Where the HOST variable is the URL sent by each participant (you), the JWT must be supplied to us (by any means) and it has to be unique per transaction.

You must comply with the following requirements:

- The microservice must be containerized and can be deployed on any machine or in the cloud.
- It is required to use a load balancer with a minimum of two nodes with the same microservice.
- The infrastructure code must be versionated.
- The pipeline should be configured as a code and needs to be stored in a repository.

Minimum requirements for the pipeline:

- Use of Dependency Management
- It should have two stages at minimum: “build” and “test”, and can have the stages that you want.
- Must be automatic and can be executed by any branch, the master branch always deploys to the production environment. If it is required you can create more environments: development or testing. Additionally you could execute the pipeline on demand and you can deploy any version of the microservice (In the case that more than one version exists)
- The pipeline should be configured as a code and needs to be stored in a repository.

The project must include:

- Automatic testing of any type.
- Static code revision.
- Dynamic grow.
- API Manager for API key and JWT

The revision is going to focus on the pipeline and we expect you to use "Clean Code" and "TDD" for the development of the microservice (which implies a required coverage that will be analyzed).